

**APPLICATION FOR
UNITED STATES PATENT
IN THE NAME**

Of

ALEJANDRO M. GRIGNETTI

For

BROWSER SYSTEM AND METHOD

DOCKET NO. 51185.00008

**Please direct communications to:
SQUIRE, SANDERS & DEMPSEY L.L.P.
600 Hansen Way
Palo Alto, CA 94304-1043
(650) 856-6500
Express Mail Number: EL701362433US**

1 BROWSER SYSTEM AND METHOD

2 Inventor: Alejandro M. Grignetti

3
4 CROSS-REFERENCE TO RELATED APPLICATION

5 This application claims priority to U.S. Provisional
6 Application No. 60/192,609, by common inventor, Sandro
7 (Alejandro) Grignetti, filed March 27, 2000, and entitled
8 "BROWSER SYSTEM AND METHOD". Application No. 60/192,609 is fully
9 incorporated herein by reference.

10
11 TECHNICAL FIELD

12 This disclosure relates to Internet appliances, and more
13 particularly but not exclusively, to portable Internet browser
14 devices that do not require connection to the Internet in order
15 to display Internet content.

16
17 BACKGROUND

18 Current devices for reading digital content include WebPAD
19 style devices, electronic books, set top boxes, WEB enabled cell
20 phones, Personal Digital Assistants (PDA's), palmtops, and laptop
21 and desktop PC (Personal Computing) devices. WebPAD style
22 devices are typically tethered to a base station via radio
23 frequency (RF), have limited or no data storage capability
24 (usually less than approximately 64 MB), use proprietary

1 operating systems (e.g., QNX, VxWorks, WinCE), and have limited
2 battery life (typically less than approximately 3 hours).
3 Additionally, WebPAD style devices are not 100% Internet
4 compliant, since these devices typically offer no support for the
5 following: (1) Macromedia ShockWave or Flash, (2) Multi-media
6 support (QuickTime, .AVI, Real Audio/Video, and the like), and
7 (3) JAVA, JavaScript, and Virtual Reality Modeling Language
8 (VRML) and other "Plug-ins". As known to those skilled in the
9 art, a plug-in is a helper application used by browsers to
10 facilitate the viewing of certain file types, or content. WebPAD
11 style devices also have limited or no viewers for .pdf, .doc,
12 .xls, .ppt, type files, and the like.

13 Electronic books are limited to a single function and have
14 no general Internet access capability. This single function uses
15 only one data format type, and this is generally a proprietary
16 format. Furthermore, electronic books have the limitation of
17 being necessarily linked to an associated proprietary content
18 database (in order to facilitate copyright protection).

19 Set top boxes, such as those provided by WebTV, require a
20 wired Internet connection in order for Internet content to be
21 viewed in the set top box. Furthermore, set top boxes are not
22 designed to be portable devices, and offer a very limited visual
23 experience due to the low resolution of a television set, and the

1 manner in which the television image is drawn on the screen
2 (i.e., interlaced format reduces resolution further).

3 WEB enabled cell phones offer a very small screen, forcing
4 WEB content to be reduced and heavily re-formatted in order to
5 fit the smaller form-factor. Web enabled Cell phones are also
6 not 100% Internet compliant, by having support only for text.

7 Palmtop devices are generally not X86 processor based (the
8 Intel 80x86 range, or compatibles from, e.g., Cyrix or Advanced
9 Micro Devices), making them incompatible with existing plug-in's
10 for viewing Internet content. Furthermore, palmtop devices
11 generally use the proprietary operating system, WinCE, which has
12 poor World Wide Web (WEB) browser support. As known to those
13 skilled in the art, a browser is an application program that
14 provides a method to look at and interact with information
15 contained on the WEB. The Web browser is typically a client
16 program that uses Hypertext Transfer Protocol (HTTP) that enables
17 the browser user to make requests to WEB servers connected to the
18 Internet. Two conventional WEB browsers that are widely used
19 today are the Netscape Navigator and the Microsoft Internet
20 Explorer.

21 Personal computers are often used for reading Internet
22 content, but they can be expensive, and non-intuitive or
23 complicated for the non-computer user. Additionally, personal
24 computers are large in size, heavy in weight, non-portable,

1 typically expensive, and may not provide reading comfort for some
2 users. Personal computers are also slow in operation, since they
3 require time to "boot up" during system start up or system reset,
4 and since they typically are processing multiple applications.
5 Additionally, it is time consuming for personal computer users to
6 access the Internet and perform browsing operations, particularly
7 when the WEB server in the Internet that they wish to access is
8 busy or when network failure occurs.

9 Current methods for browsing Internet content require the
10 existence of a continuous connection to the Internet. This
11 connection forms a physical link between the device and the
12 Internet, which limits the portability of the browsing
13 experience. In the future, when high bandwidth cellular telephone
14 data access is available, the ability to sustain a more portable
15 connection will be possible. But even in these circumstances,
16 sustaining a continuous connection to the Internet will result in
17 a high price. There will still be a need to mitigate these prices
18 even when Internet access is nearly ubiquitous.

19 Thus, there is a need for an apparatus and method that will
20 overcome the above-mentioned deficiencies of conventional methods
21 and systems. There is also a need for a portable apparatus and
22 method to view Internet content. There is also a need for a
23 portable apparatus and method that will permit a user to view and
24 browse through Internet content at a faster speed and without the

1 requirement of a continuous connection to the Internet. There is
2 also a need for a portable apparatus and method that will permit
3 a user to conveniently view and browse through Internet content
4 at any time and anywhere. There is also a need for a portable
5 apparatus and method that will allow caching of content to
6 minimize connection times. There is also a need for a portable
7 apparatus and method that will permit a user to view and browse
8 through Internet content in a convenient manner and with less
9 difficulty. There is also a need for an inexpensive and easy-to-
10 manufacture apparatus for permitting a user to view and browse
11 through Internet content.

12 13 BRIEF DESCRIPTION OF THE DRAWINGS

14 Non-limiting and non-exhaustive embodiments of the present
15 invention are described with reference to the following figures,
16 wherein like reference numerals refer to like parts throughout
17 the various views unless otherwise specified.

18 Figure 1 is a block diagram of a portable peripheral
19 browser device according to an embodiment of the invention.

20 Figure 2 is a block diagram of the bottom view of the
21 browser device 100.

22 Figure 3 is a block diagram showing the browser device for
23 connection to a cradle and to the Internet.

1 Figure 4 is a block diagram of the system components in the
2 browser device of Figure 1.

3 Figure 5 is a block diagram of the maximized screen area in
4 the browser device.

5 Figure 6 is a block diagram of the Index structure in the
6 browser device.

7 Figure 7 is a screen shot of the main contents page that
8 shows an index of links to the different content categories.

9 Figure 8 is a screen shot of a page showing the contents in
10 the newspapers category.

11 Figure 9 is screen shot of a page showing the contents in
12 the magazines category.

13 Figure 10 is a screen shot of a page showing the contents in
14 the WEB sites category.

15 Figure 11 is a screen shot of a page showing the contents in
16 the Documents category 630.

17 Figure 12 is a flowchart diagram of a method of displaying
18 and creating pictures in the browser device of Figure 1.

19 Figure 13 is a block diagram of a data casting system
20 according to an embodiment of the invention.

21
22 DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS

23 Embodiments of a method and system to provide a portable
24 browsing device for viewing content received from the Internet

1 without requiring the browser device to be connected to the
2 Internet during the viewing of the content are described herein.
3 As an overview, an embodiment of the invention provides a
4 portable browser device including: a touch sensitive LCD screen
5 capable to display images of Internet content, a non-volatile
6 storage medium coupled to the processor and capable to store
7 content received from a data communication network such as the
8 Internet, a content browsing software, a processor capable to
9 execute the content browsing software, a battery pack to power
10 the device, an I/O port and an ergonomically designed enclosure
11 to house all of these components in a portable form factor.

12 The present invention may advantageously provide a system
13 and method for enabling users to more conveniently view Internet
14 content at any time, in any place or in any manner and for
15 enabling users to access and browse Internet content at a faster
16 speed. The present invention may also advantageously provide a
17 browser device that is portable and that can be used to display
18 Internet content without requiring the browser device to be
19 connected to the Internet during the viewing of the Internet
20 content. The present invention may also advantageously provide a
21 browser device that is smaller in size, lighter in weight, lesser
22 in cost, has reduced design/production complexity, and has less
23 maintenance requirements as compared to conventional Personal
24 Computing devices. The present invention may also advantageously

1 minimize the amount of software required for Internet Browsing,
2 and this feature minimizes the number of commands required from
3 the user to operate the browser device. Thus, the present
4 invention may provide a browser device that is easy to use even
5 for non-computer users. The present invention may also
6 advantageously provide a browser device that can download and
7 store for viewing a large amount and wide variety of digital
8 content such as, for example, WEB Sites, periodicals such as "The
9 Wall Street Journal" or "Time", digital photographs, documents,
10 multimedia (e.g., movies), music, and the like.

11 In the description herein, numerous specific details are
12 provided, such as the description of system components and
13 software, to provide a thorough understanding of embodiments of
14 the invention. One skilled in the relevant art will recognize,
15 however, that the invention can be practiced without one or more
16 of the specific details, or with other methods, components,
17 materials, parts, and the like. In other instances, well-known
18 structures, materials, or operations are not shown or described
19 in detail to avoid obscuring aspects of the invention.
20 Reference throughout this specification to "one embodiment" or
21 "an embodiment" means that a particular feature, structure, or
22 characteristic described in connection with the embodiment is
23 included in at least one embodiment of the present invention.
24 Thus, the appearances of the phrases "in one embodiment" or "in

1 an embodiment" in various places throughout this specification
2 are not necessarily all referring to the same embodiment.
3 Furthermore, the particular features, structures, or
4 characteristics may be combined in any suitable manner in one or
5 more embodiments.

6 Figure 1 is a block diagram of a portable peripheral browser
7 device 100 in accordance with an embodiment of the present
8 invention. In one embodiment, the peripheral browser device 100
9 is the device known as the *ProGearTM*, which is available from
10 frontpath Incorporated, 2841 Mission College Boulevard, Santa
11 Clara, California 95054. The peripheral browser device 100 is
12 typically small in size (e.g., approximately 1/2 inch in
13 thickness and approximately 8 inches by 10 inches in surface
14 area), light in weight (e.g., approximately 2 pounds), and
15 portable.

16 In one embodiment, the peripheral browser device 100 is
17 configured to display digital content on a screen 115 in portrait
18 mode by default, but landscape mode is possible as well.
19 Portrait mode is similar to standard paper and book format, and
20 was chosen to reduce the amount of scrolling that is required
21 while reading content. For example, the digital content may be
22 hypertext markup language (HTML) and/or JAVA content that was
23 previously downloaded from a data communication network such as
24 the Internet and is stored in an on-board non-volatile storage

1 120. The digital content stored in the non-volatile storage 120
2 may also be any other type of digital content that was previously
3 downloaded from a data communication network. It is noted that
4 the Internet is chosen as an example of a data communication
5 network in this disclosure because it is a well-established
6 network, and connectivity to the Internet is easily made.
7 However, it is noted that a global communication network, such
8 as the Internet, is not required to practice other embodiments
9 of the invention. The digital content that is stored in the on-
10 board non-volatile storage 120 may be downloaded from other
11 types of data communication network such as, for example, a
12 locally provided and maintained communication network.

13 As an example, the screen 115 comprises a liquid crystal
14 display (LCD) panel 115a (Figure 4) with a screen size of
15 approximately 10.4 inches and a screen resolution of 1024 by 768.
16 Other screen sizes and resolution values may also be used for the
17 screen 115. The display technology may also vary for screen 115.
18 For example, thin film transistor (TFT) LCDs (i.e., active matrix
19 displays) or super twisted nematic (STN, or passive matrix)
20 displays may be used as the display technology for screen 115.
21 Bi-stable LCD displays, such as electronic ink (Eink), which have
22 less power requirements, may also be used for the display
23 technology for screen 115, as could Organic Light Emitting Diodes
24 (OLED's, or LEP's Light Emitting Polymers).

1 In one embodiment, the screen 115 includes a four or five
2 wire resistive touchscreen 115b (Figure 4). A power button 125
3 is used for turning the peripheral browser device 100 on or off.
4 A multifunction joystick 135 is used for accessing a menu which
5 allows access to all functionalities in the peripheral browser
6 device 100, as well as permitting the user to perform scrolling
7 functions (scrolling up, down, left, and right functions) on
8 screen 115. The joystick 135 can be in any suitable compact
9 configuration that permits ease of use for the user. A stylus
10 126 permits fine touch access to screen contents too small for
11 finger selection. A DC Input jack 127 provides connection to a
12 DC power source (not shown) for recharging a battery in the
13 browser device 100 and permits transmission of power into the
14 browser device 100 when battery power is low. The device 100
15 also includes a cover 128 to protect the LCD panel/touch screen
16 during storage, and also has a switch (cover sensor 129) to turn
17 the browser device 100 off when the cover 128 is closed (i.e.,
18 when the cover 128 is inserted into the cover latch). Other
19 features that may be included in the browser device 100 includes,
20 for example, a speaker 121, microphone 122, stylus storage area
21 123, and/or earphone jack 124.

22 Figure 2 is a block diagram of the bottom view of the
23 browser device 100. Docking connectors 131a and 131b permits the
24 browser device 100 to be connected to, for example, a cradle 207

1 (Figure 3), base station 1315 (Figure 13) or other suitable
2 docking stations. A battery compartment 159 in browser device
3 100 may house a suitable power source such as a battery.

4 In one embodiment, the browser device 100 will only run a
5 single application, an Internet browser 130 (see Figure 4). This
6 application 130 is automatically loaded when the system of
7 browser device 100 initializes, and there is no technique of
8 closing this application 100 during normal operation of the
9 browser device 100. The browser device 100 functions like a
10 hardware Internet browser, and not like a typical PC that runs
11 many applications. There is no "Desktop" to be viewed.

12 In one embodiment, the client area 500 in screen 115 is
13 maximized so that the space per content to be shown on screen 115
14 is maximized (see Figure 5). The client area 500 is maximized by
15 the use of, for example, no menu bar, no title bar, and no window
16 borders. A minimal scroll bar is used for visual aide only, to
17 show location within a document. A menu 505 (Figure 5) can be
18 brought up with the use of the multi function joystick device 135
19 (Figure 1). The only application that the user sees on the
20 screen 115 is the client window of the browser 150 (Figure 4),
21 and, as a result, the screen 115 advantageously has maximum space
22 leftover for displaying Internet content.

23 When disconnected from an Internet connection, the
24 peripheral browser device 100 performs the browsing functions

1 that are similarly performed by an Internet WEB browser, except
2 that the peripheral browser device 100 accesses digital content
3 stored in non-volatile storage 120 in order to display digital
4 content on screen 115. The digital content stored in non-
5 volatile storage 120 was previously downloaded from the Internet.
6 The non-volatile storage 120 is, for example, a hard disk drive.
7 However, other types of storage devices, such as Micro Drives
8 available from, for example, International Business Machines
9 Corporation, Compact Flash, or Magnetic Random Access Memory
10 (MRAM) may also be used for the non-volatile storage 120. The
11 non-volatile storage 120 preferably has sufficient size so that
12 it can store WEB sites, documents, electronic magazines, or other
13 types of periodicals or digital content. For example, the size
14 of non-volatile storage 120 can be approximately 340 to 1000
15 megabytes. Therefore, the peripheral browser device 100
16 advantageously permits a user to view and browse through Internet
17 content stored in non-volatile storage 120, at any time and any
18 where, without requiring a connection to the Internet when the
19 user is viewing the Internet content.

20 The peripheral browser device 100 is typically an accessory
21 device that can be used with a host-processing device 209 such as
22 a desktop personal computer. The host-processing device is used
23 for connection to the Internet and for permitting a user to
24 performing printing commands or archiving functions. Thus, the

1 user can direct the host-processing device to download (or
2 electronically store or "print") Internet content to non-volatile
3 storage 120 of peripheral browser device 100, instead of
4 conventionally printing the Internet content in a paper printing
5 device.

6 Alternatively, instead of being connected to a host
7 processor 209, the peripheral browser device 100 can be
8 connected directly to the Internet via an analog or digital
9 modem so that Internet content can be downloaded to the non-
10 volatile storage 120. The device can also be configured to use
11 a standard network or a RF (Radio Frequency) network connection
12 to access the Internet for content.

13 The peripheral browser device 100 may advantageously provide
14 an advancement in the field of electronic publishing by
15 facilitating the viewing and consumption of content without the
16 need for traditional paper based printing of the content.

17 Electronic subscription copies of periodicals, such as newspapers
18 and/or magazines or other files, can be downloaded from the
19 Internet to the peripheral browser device 100. Alternatively,
20 electronic subscription copies of books or other files may have
21 been received in the host-processing device via e-mail, and these
22 electronic copies can then be downloaded to the peripheral
23 browser device 100. Additionally, various other types of digital
24 content can be downloaded to and/or stored in the non-volatile

1 storage 120 of the peripheral browser device 100 and viewed by
2 the user on the screen 115. For example, the user can store and
3 view digital photographs, digital documents, and/or other digital
4 content. Commuters, travelers, and readers can greatly benefit
5 by using the peripheral browser device 100 because of its
6 portability. By use of the peripheral browser device 100, such
7 persons will be able to download, store, browse through, and/or
8 read digital content from the Internet, at any time and anywhere
9 and without the constraint of requiring an Internet connection
10 while viewing such digital content.

11 The peripheral browser device 100 may also be beneficial in
12 the area of education. Recent studies have shown that use of
13 laptop or notebook computers in the classrooms can improve the
14 learning process for students. The peripheral browser device 100
15 can offer many of the advantages of a laptop or notebook computer
16 for educational purposes, but at a reduced cost. For example,
17 classroom teaching materials can be designed as HTML documents
18 with JAVA content. These teaching materials can be designed for
19 teaching students as well as for monitoring the progress of
20 students in the classroom or at home. In particular, reference
21 books, text books, lessons, classroom or homework assignments,
22 and/or exams can be downloaded to the peripheral browser device
23 100 from a teacher's desktop computer. The students can then
24 access these downloaded materials in the peripheral browser

1 devices 100, whether the students are in classroom or at home.
2 Teachers will benefit from the improved ability to tailor their
3 educational materials for each student. A student benefits by
4 having just one peripheral browser device that can be used for
5 all his/her courses and textbooks so that there is no need for
6 him/her to carry multiple heavy books between classes.
7 Additionally, students can reduce the cost of books for their
8 courses.

9 The peripheral browser device 100 is also beneficial for
10 publishers and for subscribers. Users can now have available in
11 their peripheral browser devices 100 their favorite subscriptions
12 of magazines, newspapers, and/or other periodicals which have
13 been downloaded and stored in non-volatile storage 120 of the
14 peripheral browser device 100. The electronic publishing feature
15 enabled by the peripheral browser device 100 permits publishers
16 to reduce or eliminate costs in paper printing/publishing and in
17 distribution, without negatively impacting the readership of the
18 periodicals or the value of the advertisements in the
19 periodicals. Indeed, advertisements can be made even more
20 compelling to the reader because of the ability to combine
21 traditional print based advertising with television style
22 multimedia advertising, and with Internet based (targeted)
23 advertising, a concept hereinafter called "Deep Advertising".

1 Other advantageous uses for the peripheral browser device
2 100 may include the following: a photo album for storing and
3 viewing picture images, a coffee table book, a simple game
4 machine, an electronic substitute for paper in the office or for
5 hand-outs in seminars, an Internet shopping catalog, a global
6 positioning system (GPS) guided map, a reference for technicians
7 or engineers building or designing complicated structures or
8 devices (e.g., airplanes by the Boeing Company), a digital
9 notepad and reference guide for doctors in hospital settings, a
10 device for storing an electronic copy of books or pamphlets, and
11 numerous other uses and applications.

12 Other particular advantages of the peripheral browser device
13 100 include its reduced size, weight, and design/manufacturing
14 complexity, and its improved portability and ease of use, as
15 compared to conventional computers.

16 In one embodiment, the browser device 100 is an Instant-On
17 device. Instant-On is an established method of implementing a
18 system that initializes "instantly", by effectively minimizing
19 power consumption in a pseudo off state. When the system is
20 turned off, the system of device 100 merely goes into an
21 extremely low power state that maintains the state of DRAM. This
22 DRAM state is known as self refresh mode, and as known to those
23 skilled in the art, self refresh DRAM memory is widely available.
24 When the user turns the system back on, the system is already

1 initialized, and needs to merely restore power to system
2 components to achieve full functionality. By allowing the system
3 to initialize instantly, the browser device 100 is more
4 accessible to users.

5 Figure 4 is a block diagram illustrating the important
6 components of the peripheral browser device 100. Additionally,
7 some or all of various components shown in Figure 4 may be
8 integrated in an application specific integrated circuit (ASIC)
9 or field programmable gate array (FPGA) to help reduce cost and
10 lessen manufacturing difficulties. Other types of assemblies and
11 manufacturing techniques may also be used for assembling the
12 components shown in Figure 4.

13 A processor 200 performs the control functions of the
14 peripheral browser device 100. The LCD display screen 115a will
15 show the contents of the non-volatile storage (Internet content)
16 and the touch screen 115b will allow navigation of the content
17 similarly to a conventional PC, in that selection of hypertext
18 links accomplishes movement within the stored content. The
19 processor 100 runs an Operating System 131 and the Internet
20 browser (content browser engine) 130, and associated peripheral
21 applications or helper applications 133 that the browser 130 may
22 require while decoding WEB pages. The browser device 100 is
23 configured to run just the browser application 130, such that it
24 is not possible to shut down the browser application 130 to get

1 to a "desktop" state. The browser 130 can, as needed, call upon
2 helper applications (helper engines) 132 to assist in the
3 decoding of certain types of Internet content.

4 When the peripheral browser device 100 is connected to a
5 host-processing device 209 such as a computer, the following
6 unique functions are also advantageously permissible. A device
7 driver 210 resident in the host computer 209 permits the
8 peripheral browser device 100 to be seen as an optional printing
9 device in that computer 209. This allows users of the host-
10 processing device 209 to "print" any content to the peripheral
11 browser device 100 that would otherwise be printed on a paper if
12 a conventional printer device were used instead. When printing
13 to the peripheral browser device 100, the user simply selects the
14 peripheral browser device 100 from the list of available printer
15 devices, and the host processor resident device driver 210
16 converts the selected document into HTML format for subsequent
17 download to the peripheral browser device 100. This, in effect,
18 makes the peripheral browser device 100 a host processor
19 compatible printer.

20 The peripheral browser device 100 can also advantageously
21 function as a "computer doubler" when the peripheral browser
22 device 100 is connected to the host processing device 209. For
23 example, the user can use the peripheral browser device 100 to
24 surf the World Wide Web by use of a connection (via the host

1 processing device 209) to the Internet, while a second user can
2 simultaneously use the host-processing device 209 for other tasks
3 such as playing a computer video game or performing word
4 processing. This is accomplished by a device driver 211 in the
5 host processing device 209 where the device driver that routes
6 Transmission Control Protocol/Internet Protocol (TCP/IP) requests
7 from the peripheral browser device 100 to the established
8 Internet connection, with minimal host CPU overhead. In typical
9 multitasking operating systems (such as with Microsoft Windows)
10 the CPU overhead required for servicing TCP/IP handoff such as
11 the above is so small as to allow nearly transparent processing
12 of the TCP/IP requests, while devoting the majority of the CPU to
13 the other processes that the host CPU may be performing.

14 Additionally, the peripheral browser device 100 permits the
15 scheduled downloads of digital content from the Internet during
16 specified time periods. For example, the digital content from
17 the Internet may be downloaded at night to the non-volatile
18 storage 120, and the user may conveniently view the downloaded
19 content or browse through the downloaded files the following day.
20 This is accomplished by using a helper application 133 on the
21 peripheral browser device 100, and a device driver 212 on the
22 host-processing device 209. The helper application 133 on the
23 peripheral browser device 100 will wake the peripheral browser
24 device 100 at the specified time, and begin making download

1 requests from the Internet connection. Content that had
2 previously been marked for download or refresh by the user would
3 be requested at this time. The host processing device driver 212
4 will take these requests from the helper application 133 and
5 route them as if a user were making the download requests
6 directly from the host processing device 209. The host resident
7 device driver 212 makes no distinction between user requests made
8 directly from the host processing device 209 or unattended
9 automatic requests sent from the helper application 133.

10 The input/output (I/O) port 205 permits the peripheral
11 browser device 100 to be connected to an Internet connection 206
12 (Figure 3). This Internet connection 206 may be, for example,
13 through a host computer, a network, or a modem. A physical
14 network connection may be through Ethernet, Radio Frequency (RF),
15 or Universal Serial Bus (USB).

16 The peripheral browser device 100 is powered by a power
17 source that is, for example, a battery pack. The power source is
18 recharged when the peripheral browser device 100 is connected to
19 an AC/DC converter, or when the peripheral browser device is
20 placed in its base station (or cradle) 207 (Figure 3).

21 In one embodiment, the system components are all stored in a
22 special closed area of the non-volatile storage 120. The system
23 components include, the operating system 131, the browser 130, a
24 server 136, device drivers 160, and helper applications 133. The

1 system components are typically implemented as software (i.e.,
2 programs, instructions, modules, engines or the like). The
3 operating system 131 may be, for example, the Linux operating
4 system. The browser 130 may be, for example, a Netscape (or
5 Mozilla) browser. As known to those skilled in the art, Mozilla
6 is a public domain WEB Browser. The source code to Mozilla is
7 available for modification, free of charge. Modification of this
8 browser source code allows one skilled in the art to incorporate
9 some of the functionality of a WEB server into the Mozilla
10 browser code base, and this reduces the overall size of the
11 software package.

12 The WEB server 136 controls or filters the TCP/IP packets
13 between the browser 130 of the peripheral browser device 100 and
14 the Internet content (either in non-volatile storage 120, or on
15 the "live" Internet). The server 136 launches appropriate helper
16 applications 133, and selects the location of content, either
17 locally from non-volatile storage 120, or from the Internet.
18 Because the peripheral browser device 100 is designed primarily
19 for offline browsing, the server is placed in the off-line mode
20 (requesting content from the non-volatile storage 120) by
21 default. When a refresh or update is requested (either by the
22 user, or by the auto update utility), the server 136 is switched
23 to live content mode, where content is requested from the
24 Internet connected to the I/O port. The user may also select a

1 live connection for browsing as well, if, for example, an RF link
2 to the Internet is present continuously. This mode of operation
3 can then allow the peripheral browser device 100 to function as a
4 live wireless browser of Internet content.

5 The browser 130 may also include plug-ins to assist in the
6 viewing of files that are not in HTML format. These files may,
7 for example, be Acrobat files (.pdf), photographs (.jpg, .gif,
8 .tiff, .bmp), multimedia (RealAudio/Video, QuickTime, MPEG) or
9 sound (.wav, or .MP3). The plug-ins are similar to helper
10 applications 133, in that the plug-ins allow the browser 130 to
11 view files that are not .html format. Plug-ins differ from
12 other helper applications in that the plug-ins use a preferred
13 communication to the browser 130 allowing the plug-in to
14 function from within the browser. In some instances, this is
15 preferred, when, for example, a .pdf file may be viewed from
16 within the browser 130 window, rather than in a separate
17 application window. However, the added work of that level of
18 integration is not always justified, so the peripheral browser
19 device 100 has both types of helper applications in one
20 embodiment.

21 Other elements in the system shown in Figure 4 are also now
22 described. A graphics controller 150 handles the formation of
23 graphic elements on the LCD 115a. The graphics controller 150
24 communicates with the operating system 131, and takes the

1 drawing commands from the browser application 130 and converts
2 them to pixels on the LCD 115a. The touch controller 152
3 converts the analog signals of the touch screen 115b into
4 digital point coordinates used by the operating system 131 as
5 user (or mouse) input. The joystick controller 153 converts the
6 analog signals of the joystick 135 into digital commands for the
7 operating system 131 to interpret as user commands (such as
8 scrolling). The device 100 subsystems are started by the ROM
9 BIOS software, which initializes all hardware (HW) components,
10 and then loads the operating system 131 to the DRAM memory 154.
11 The processor 200 and operating system 131 then begin executing
12 programs, such as the browser application 130, to make the
13 device 100 perform its functions. The DRAM memory 154 is where
14 the processor 200 executes all software instructions. As known
15 to those skilled in the art, processors require software to
16 perform any functions, and this software requires memory to
17 execute within. The system bus 151 is the primary avenue for
18 communications of all devices with the DRAM memory and the
19 processor. The non-volatile storage 120 has a closed area
20 within it to store all system software. The operating system
21 131, the browser application 130, and all helper applications
22 133, and device drivers 160 are placed here for permanent
23 storage.

24

1 Content Pointer Database

2 The peripheral browser device 100 obtains content from the
3 Internet. This content is downloaded to the browser device 100
4 by use of an automated download utility 137. The automated
5 download utility 137 has preset content starting points that are
6 simply hyperlinks (similar to the links stored in a Bookmark or
7 favorites file), and uses general instructions to download
8 certain depths of content from the user selected sites in the
9 Internet. These general instructions (which have variables that
10 control it, and that are set by the user) control the manner in
11 which a WEB site is traversed and copied to the local non-
12 volatile storage 120 for caching. Starting from an initial web
13 page, the program will download all web pages that are
14 referenced within that initial web page. The program will also
15 translate all links in the web page, from absolute Internet
16 addresses, to relative file system type addresses. For example,
17 a file residing on the Internet, and having a link to the
18 absolute address `HTTP://www.frontpath.com/progear.html`, will
19 have the link translated to
20 `FILE:///c:/webcontent/www.frontpath.com/progear.html`. As links
21 are traversed, copied and translated, the automated download
22 utility program 137 will count the depth of sites (depth is a
23 representation of how many links deep the program will traverse.
24 If the program is set to traverse to a depth of 2 links deep, a

1 web page has two links, each of which has two links, will result
2 in a total of 4 web pages being stored).

3 The automated download utility 137 will cache in non-
4 volatile storage 120 all of the requested content, and make
5 entries in the content pointer database 140 to indicate
6 appropriate information about each downloaded WEB site. The
7 information recorded includes, for example, the starting point
8 location of the content within the non-volatile storage 120, the
9 date and time of the download of the content, any associated
10 icons that represent the downloaded content visually, and an
11 indication of the size of the downloaded content pointed to by
12 the start link. Also included in the database entry is a linked
13 list of all disk sectors and file names that the link tree
14 created to build that download (all sites traversed, however
15 many links deep it went). This last entry facilitates the
16 removal of content when disk-reuse (described below) is
17 initiated.

18 Inevitably however, the auto download utility will download
19 content that was not required, or it will not download critical
20 content, since it is impossible for a remote program to
21 understand the content. This is due to the fact that a simple
22 depth variable is insufficient to understand the format of every
23 article written. For example, some publishers may wish to get
24 as much advertising into an article as possible, so a 10 page

1 article may be divided into 20 linked pages, each with a
2 different add. Other publishers, if they are subscription
3 rather than advertising based, may place the article on a single
4 long web page, to facilitate reading. A simple program cannot
5 take the two scenarios into account; therefore the concept of
6 content bundles is introduced. Content bundles are to the web,
7 what books, newspapers and magazines are to paper: groups of
8 like content bound together in one package. Content bundles are
9 groups of self-referencing web pages that are put together by
10 the publisher of that content, and then compressed for ease of
11 delivery. The content bundles solve the problem of incomplete or
12 extra content created by the automatic download utility 137.
13 The content bundle represents a complete image of a magazine,
14 newspaper or even a web-based catalog (or any web based
15 content), all wrapped up neatly into an easy to deliver bundle.
16 Content bundles can also be downloaded by the automatic download
17 utility 137; they simply do not require any depth control (a
18 content bundle is one link deep).

19 The peripheral browser device 100 accesses all stored
20 content by use of the main contents page 600 (index.html in
21 Figure 6). This main contents page 600 is a list of pre-
22 formatted links and icons that link the user to automatically
23 generated pages for each content type. These Content categories
24 are general groups of content bound by a particular source type.

1 Initial content types may be, for example, Newspapers 605,
2 Magazines 610, Books 615, WEB sites 620, Photographs 625,
3 Documents 630, and Reference 635. These are static names for
4 dynamic pages created during content loading from the content
5 pointer database. These content types are displayed in the
6 default start-up page (i.e., the main contents page 600) for the
7 peripheral browser device 100.

8 Figure 7 is a screen shot view of the main contents page
9 600 that shows an index of links to the different content
10 categories such as newspapers 605, magazines 610, WEB sites 620,
11 documents 630, and photographs 625. When an individual content
12 category is selected, all content of that type is arranged on a
13 page for selection by the user. This page is created by a
14 Common Gateway Interface (CGI) script that reads the content
15 pointer database 140 and arranges start points from the database
16 140 into a WEB page of links for that content type. The CGI
17 script only creates the page in response to a change in the
18 content database 140, so static content will not cause a
19 regeneration of the, for example, books page, unless there are
20 recent downloads of the book content type. These web pages are
21 inter-linked to allow simple forwards and backwards browsing
22 among the content categories (as shown in the figures).
23 Selection of particular content will cause the browser 130 to
24 jump to the location of that content on the non-volatile storage

1 120, and subsequent links come from the tree of content that that
2 particular content has stored in non-volatile storage 120. Any
3 content that has been placed in non-volatile storage 120 in
4 content bundle format runs through the added step of on-the-fly
5 decompression prior to display, so as to keep the content in
6 compressed form. As known to those skilled in the art, a CGI
7 script is a small program that runs on an application that is
8 executed by an HTTP server machine in response to a request by a
9 client, such as a WEB browser.

10 Figure 8 is a page screen shot showing the contents in the
11 newspapers category 605. For example, the contents include links
12 to the Boston Globe™, Washington Post™, and the Mercury Center
13 of the San Jose Mercury News™.

14 Figure 9 is page screen shot showing the contents in the
15 magazines category 610. For example, the contents include links
16 to the Time™, PC Magazine™, PC Computing™, and other links.

17 Figure 10 is a page screen shot showing the contents in the
18 WEB sites category 620. For example, the contents include links
19 to the WEB sites of "4th Wave™," "The Nine Planets™," and other
20 WEB site links.

21 Figure 11 is a page screen shot showing the contents in the
22 Documents category 630. For example, the contents include links

1 to "The Geek Gourmet," "Beatles Album Lyrics," and other
2 documents and electronic books.

3 The content pointer database 140 is a simple database
4 containing entries for each type of content stored in the non-
5 volatile storage 120. In one embodiment, the content pointer
6 database 140 is an area in memory of the non-volatile storage
7 120. These entries contain information regarding the location
8 and usage of a content. The content pointer database is created
9 and updated by the auto download utility application that places
10 content in non-volatile storage 120. The database 140 contains
11 one entry for each content start point, not for each individual
12 piece of content. Additionally, the database 140 contains a
13 linked list of all the files that comprise that content tree,
14 for use in helping to clean the disk after the content is
15 selected for removal. There are certain properties that can be
16 given to pieces of content within any content type, or below any
17 start point, which can cause the creation of different and new
18 content pointers in the database. Precious content identifies
19 content that is not allowed to be erased unless explicitly
20 directed by the user. Precious content is designated by the
21 user in certain categories, and automatically in others. All
22 photographs, books, documents and reference works are
23 automatically assigned the precious content attribute, therefore
24 deletion requires user intervention. Users may select

1 individual pages of newspaper, magazine or web site type content
2 for more permanent storage. Users select the precious attribute
3 from the pop-up menu that can be selected with the joystick when
4 that page displayed. Content that is in content bundle format
5 when it is given a precious attribute is copied out of the
6 content bundle in uncompressed format, and saved in the non-
7 volatile storage. This allows content bundles to be deleted as
8 a whole.

9 In one embodiment, the precious attribute is utilized in
10 conjunction with the automatic disk reuse program 141. This is
11 simply a garbage collection type disk cache cleanup that uses a
12 Least Recently Used (LRU) algorithm to determine when blocks of
13 non-volatile storage 120 are to be deleted for use by new
14 content. The LRU scheme works in conjunction with the content
15 pointer database 140 to identify content that is "stale", or has
16 not been accessed in a user determined amount of time, and that
17 does not have a precious attribute. Content that is, for
18 example, more than one month old (this time period is user
19 selectable for each content type), may be selected for automatic
20 deletion. This will make room for the storage of newer content.
21 Contents are deleted from the Content pointer database start
22 point, and down the link tree from there. The link traversal
23 during deletion is facilitated by the link tree created during .
24 the initial content write to non-volatile storage. The link

1 tree allows the cleanup utility (LRU) to simply delete a series
2 of files that are known to be part of a particular initial start
3 point, or content type. This reduces or eliminates the
4 possibility of Dead Links left in non-volatile storage. Dead
5 links are pages of WEB content that have no method of being
6 accessed by either a WEB page link, or a content pointer
7 database link. The LRU scheme is made easier with Content
8 Bundles, in that large areas of storage are freed with a single
9 file deletion. Each content type can have different default
10 staleness times, so, for example, a newspaper entry may be stale
11 after three or four days, while a magazine may not become stale
12 for four months. These staleness attributes are also stored in
13 the content pointer database, and are user changeable.

14 In one embodiment, the Disk Reuse Algorithm (DRA) 141 will
15 only need to run when the Automated Download Utility 137 is also
16 running. The DRA 141 will run just prior and then during
17 automated downloading to free up non-volatile storage space as
18 needed for the content download. As content is removed, the
19 associated content pointer in the content pointer database is
20 also cleared. Any content marked as precious is retained, its
21 content pointer having been created at the time that the content
22 was made precious. Only content that has no content pointer,
23 and is "below" or only referenced by a to-be-deleted content
24 pointer, is removed automatically.

1 The automatic download utility 137 is a type of wake-up
2 alarm for the device. To update the pad requires an Internet
3 connection present, so it is assumed that the device is either
4 connected (either wired or wirelessly) to the Internet during
5 system storage (overnight for example). At the user-selected
6 time, the automatic download utility 137 will wake up the device
7 (turn on power to system components, except the sound and
8 graphics sub-systems) and initiated a refresh of content from
9 the Internet. The utility 137 will simultaneously run the DRA
10 141 to free space on the non-volatile storage 120 as needed for
11 content. Deletion of content by the DRA 141 is atomic, i.e.
12 disk space is freed up in increments of content pointer heads,
13 or trees, so chunks are freed at a time, not individual files or
14 bytes.

16 Picture Viewer Database Application

17 In some circumstances, the server 136 will perform more
18 than just delivering HTML content pages for viewing on screen
19 115. For example, in Figure 12, assume that there is a request
20 (300) from the user to display pictures stored in a directory in
21 the non-volatile storage 120. This request is passed (302) to
22 the server 136. The server 136 then reads the request and then
23 looks into the directory in non-volatile storage 120 to determine
24 (304) the types of files stored in the directory. When the

1 server 136 determines that the files in the directory are all
2 picture files, the server 136 informs (306) the browser 130 that
3 the request will be handled by another application (i.e., the
4 picture viewer 143). Instead of displaying a directory listing
5 using ftp format because the content location in the local hard
6 drive has no HTML files, the server 136 runs a helper
7 application 133 designed to display (308) the pictures in the
8 directory in a user-friendly manner. This resolves several
9 common problems with the typical method of displaying pictures
10 in a browser. Normally, the browser will display a view of the
11 files in the directory, and each picture will be a link which,
12 when selected, displays the picture in the browser. To view the
13 next picture, the user goes back with the browser, and selects
14 the next picture, and so on. This previous technique is
15 cumbersome and impractical for simple picture viewing.

16 Furthermore, when pictures are stored in the hard drive in
17 the first place, users are forced to create a subdirectory for
18 the storage of the new pictures. This is simply due to the need
19 to organize the pictures in some manner, much like people will
20 save pictures based upon the roll of film that they were
21 originally shot with. Then, to view the digital pictures, the
22 user enters the directory that contains the pictures from that
23 roll of film, and views a slide show of that roll. However, if
24 there are two or more different slides shows that a picture

1 could be included in, the individual picture must be copied to
2 each individual directory, thus duplicating files.

3 The new digital viewer application 143 will solve both of
4 these problems in a unique and simple manner. All of the
5 pictures stored in the non-volatile storage 120 will remain in a
6 single directory, e.g., a storage place for an entire family's
7 photographs. Viewing of individual pictures happens through
8 user created albums, which are just lists of individual pictures
9 that are to be viewed together, one after the other. The
10 picture location, name and other data are stored in a picture
11 viewer database 144. Each picture, when originally entered into
12 the picture viewer database 144 is assigned the "default" name
13 of the roll of film (or initial event name) that would otherwise
14 correspond to the initial directory for the pictures. This
15 represents the default picture album with which the pictures may
16 be viewed from. Subsequent albums (created by the user to view
17 the same picture from) indicate the second, third, fourth, or
18 more preferred viewing lists for the same picture.

19 For example, a family takes pictures during a vacation, a
20 birthday party, and a graduation. These pictures are initially
21 placed in the default categories "Family Vacation", "Brianna's
22 Birthday", and "Brianna's Graduation". These default categories
23 become the default album for viewing these pictures. However,
24 all the pictures are placed in one directory together, and their

1 filenames are irrelevant (they could just be the number sequence
2 that the digital camera gave them, e.g.: DSC000191.JPG to
3 DSC000217.JPG). The picture viewer database 144 contains
4 entries for each picture, with default album entries and file
5 name entries filled in. Each database entry contains
6 information about one picture including: original file name,
7 filename in storage cache, Default Album name, Album Name list
8 (with, e.g., storage for 250 album names), file creation date,
9 last file change date. The files for the pictures are not
10 changed or modified in any way, so as to retain their inherent
11 flexibility and portability. Only the database entries are
12 modified, and only by the picture viewer application 143.

13 Then, assume the family wants to create a new view of the
14 existing pictures in the picture database, this time including
15 only pictures of Brianna. This is handled by creating a new
16 album of all the Brianna pictures. With this method according
17 to an embodiment of the invention, the process is simple, as the
18 default albums are viewed, if a picture looks good for another
19 album, the user selects the New Album menu item, and types in
20 the name of a new album, in this case "Pictures of Brianna".
21 This entry then forms (310) a new album, which is then added to
22 the database entry for each picture that is desired in that
23 album. The pictures themselves are not copied or moved, there
24 is just a new entry in the picture viewer database 144 for that

1 picture that identifies a new album entry in the initial album
2 view.

3 When viewing the pictures, the application creates a
4 default view that contains the names of all the current albums
5 (customizable in "Default", "Custom" and "All" categories),
6 making each album a link to the start of that albums slide show.
7 If the slide show is interrupted by a menu click (on the scroll
8 button), then a menu pops up allowing the user to place the
9 album in overview mode. This mode shows the album's individual
10 pictures in preview mode, listing as many as can fit on the
11 screen in columns and row format. This then allows the user to
12 drag and drop the pictures into the desired order for that
13 album.

14 The picture viewer database application 143 initially shows
15 a list of available albums for viewing. When an album is
16 selected, the viewer application 143 launches into the first
17 picture in that album. Pressing the forward or back browser
18 keys will move the picture to the next picture or back. The
19 picture viewer application 143 also enlarges or reduces the
20 picture to the size of the display screen, so that the maximum
21 area of the LCD is used to display each photograph. Since the
22 viewer is designed for use on this peripheral browser device
23 100, pictures are never rotated, they are displayed in the
24 manner that best fits the screen. That is, a portrait mode

1 picture is never fitted to a landscape mode screen, it is
2 displayed portrait mode, forcing the user to turn the peripheral
3 browser device 100 to view the picture, if necessary. This
4 maximizes the size of every picture, and allows full use of the
5 display screen.

6 Thus, the picture viewer database application 143 permits
7 the simplified viewing of pictures or digital slide shows for
8 the user of the peripheral browser device 100. The application
9 143 also allows users to create multiple albums without
10 duplicating files in the local storage 120. The application 143
11 also simplifies the naming requirements of files, and can
12 automatically rename files that have duplicate file names
13 without user intervention. The application 143 also allows the
14 storage of files with the minimum amount of storage space.

15 16 HDTV Connection

17 An embodiment of the present invention also may
18 advantageously avoid the necessity of setting up an Internet
19 Service Provider (ISP) account. In the high-definition
20 television (HDTV) frequency spectrum, bandwidth has been
21 allocated by the U.S. Federal government to broadcast stations
22 across the country. As known to those skilled in the art, HDTV
23 is a digital form of transmitting television images, which
24 results in a higher resolution and cleaner picture than current

1 televisions based on the National Television Standards Committee
2 (NTSC) standard. However, this new standard requires
3 significantly more bandwidth than regular television, thus
4 resulting in the allocation of new spectrum to the television
5 stations, specifically for use in HDTV transmission. This
6 allocated bandwidth is not being used fully by the broadcast
7 stations because of the lack of viewers, which in turn is due,
8 in part, to the high cost of the television sets that are
9 required. HDTV broadcasting represents a significant cost to
10 televisions stations, and the poor adoption rate leaves them
11 little chance of revenue generation to recoup these start-up
12 costs, as advertisers pay advertising fees based on the number
13 of viewers. This means that HDTV broadcasting stations could
14 benefit from other methods of revenue generation from the
15 allocated bandwidth.

16 Data casting may be used to take advantage of the unused
17 allocated bandwidth for HDTV. As known to those skilled in the
18 art, data casting involves the transmission of various kinds of
19 data as a secondary service on digital broadcasting networks.
20 The data can be information, interactive multimedia (including
21 video), or Internet content.

22 Data casting on the unused portions of the HDTV spectrum of
23 any one particular channel permits the transmission of up to
24 approximately 75 gigabytes of data per day. Referring now to

1 Figure 13, there is shown a data casting transmission system
2 1300 in accordance with an embodiment of the invention. The
3 peripheral browser device 100 has an optional base station 1305
4 that accepts Internet content from the HDTV signal 1310, for
5 storage in the non-volatile storage 120. In essence, the HDTV
6 receiver 1315 in the base station 1305 becomes a one-way
7 Internet connection.

8 As content is broadcast, the peripheral browser device 100
9 filters the received content, by use of filter software
10 (filtering engine) 1325, placing only the content bundles that
11 the user has requested into non-volatile storage 120.

12 Filtering is handled by downloading the content
13 continuously as it is broadcast. At the start of each content
14 bundle, a header 1335 describes, in one embodiment, the content
15 1336, the completed size 1337 of the bundle, a Cyclic Redundancy
16 Check (CRC) number 1338, and a content bundle date code 1339.
17 This header 1335 is used to determine the suitability of the
18 content for that particular user. If the content is not
19 required, the content is allowed to pass. If the content is
20 requested by the user, and the content is newer than the content
21 stored in non-volatile storage 120, then the content is copied
22 in its entirety to the non-volatile storage 120 of the
23 peripheral browser device 100.

1 The data cast transmission is transmitted from a
2 broadcaster transmitter 1301. An optional feedback line 1302
3 may be made available to permit the user of the browser device
4 100 to send feedback information 1303 to a server 1340 that is
5 controlled by a broadcaster. This information 1303 may be, for
6 example, usage history of a user's navigation through the
7 content bundle, including information on advertisements viewed,
8 and articles read. This information 1303 is then used to tailor
9 advertising in the future content. The server 1340 can access
10 this database 1312 for various information.

11 This embodiment allows HDTV broadcasters to benefit by
12 selling content distribution to traditional print media. This
13 distribution can augment or replace traditional printing and
14 delivery (mailing) methods, thus reducing the overall cost of
15 distributing content to end-users.

16 It is also within the scope of the present invention to
17 implement a program or code that can be stored in an
18 electronically-readable medium to permit a computer to perform
19 any of the methods described above.

20 Further, at least some of the components of this invention
21 may be implemented by using a programmed general-purpose digital
22 computer, by using application specific integrated circuits or
23 field programmable gate arrays, or by using a network of

1 interconnected components and circuits. Connections may be
2 wired, wireless, by modem, and the like.

3 The above description of illustrated embodiments of the
4 invention, including what is described in the Abstract, is not
5 intended to be exhaustive or to limit the invention to the
6 precise forms disclosed. While specific embodiments of, and
7 examples for, the invention are described herein for
8 illustrative purposes, various equivalent modifications are
9 possible within the scope of the invention, as those skilled in
10 the relevant art will recognize.

11 These modifications can be made to the invention in light
12 of the above detailed description. The terms used in the
13 following claims should not be construed to limit the invention
14 to the specific embodiments disclosed in the specification and
15 the claims. Rather, the scope of the invention is to be
16 determined entirely by the following claims, which are to be
17 construed in accordance with established doctrines of claim
18 interpretation.

19